BAB 2
LANDASAN TEORI

2.1 Peringkasan Teks Otomatis

Peringkasan teks otomatis adalah proses mengurangi teks pada dokumen menggunakan sebuah *program* komputer yang bertujuan untuk membuat ringkasan yang di dalamnya berisikan poin – poin penting dimana hasil ringkasan nantinya tidak lebih dari setengah dari dokumen asli [2].

Terdapat dua pendekatan dalam melakukan proses peringkasan teks, yaitu dengan menggunakan metode ekstraktif dan abstraktif. Metode ekstraktif adalah teknik penyusunan kalimat dengan mengambil kalimat – kalimat penting yang terdapat pada dokumen asli dan menggabungkannya menjadi dokumen yang lebih pendek. Sedangkan, teknik abstraksi adalah teknik penyusunan kalimat dengan cara mengambil kalimat – kalimat penting pada dokumen asli, kemudian membuatnya dalam bentuk kalimat lain untuk dijadikan ringkasan. Metode ekstraktif memuat beberapa bagian seperti kalimat, frasa, yang terdiri dari potongan – potongan teks yang akan membentuk sebuah ringkasan. Oleh karena itu, mengidentifikasi kalimat yang tepat untuk peringkasan merupakan faktor yang paling penting dalam metode ekstraktif [3].

2.2 *Preprocessing*

Preprocessing adalah sebuah tahapan yang bisa membuat teks menjadi data yang bisa diolah pada proses berikutnya. *Preprocessing* pada teks memiliki tujuan yaitu sebagai normalisasi kata – kata ke dalam teks serta pengurangan kosa kata yang akan diproses untuk mempermudah proses pada peringkasan teks [2].

Pada *preprocessing* terdiri dari beberapa tahap antara lain *cleaning, case folding, stopword,* dan *stemming*.

2.2.1 *Splitting*

Splitting atau segmentasi (pemecahan) adalah suatu proses pemecahan dokumen menjadi sebuah kalimat yang didasari berdasarkan tanda pemecah

kalimat. Setiap dokumen yang telah melewati proses pemecahan, akan dimasukkan ke dalam daftar kalimat. Hasil keluaran segmentasi nantinya akan berupa kumpulan kalimat yang akan digunakan pada tahap berikutnya [14]. Berikut merupakan contoh dari pemecahan kalimat yang bisa dilihat pada tabel 2.1.

Tabel 2.1 Contoh Splitting

Teks	Hasil Splitting
Salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini, disebabkan oleh perubahan dinamika kebutuhan masyarakan serta rendahnya tingkat supply akan barang. Sehingga, permintaan akan melambung tinggi dengan persediaan yang terbatas.	Salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini, disebabkan oleh perubahan dinamika kebutuhan masyarakan serta rendahnya tingkat supply akan barang.
	Sehingga, permintaan akan melambung tinggi dengan persediaan yang terbatas.

2.2.2 *Cleaning*

Tahap pembersihan karakter – karakter yang tidak diperlukan dalam proses selanjutnya. Pada tahap ini, karakter – karakter yang tidak diperlukan seperti angka dan simbol akan dihapus. Berikut merupakan contoh dari *cleaning* yang akan ditampilkan pada tabel 2.2 dibawah ini.

Tabel 2.2 Contoh Cleaning

Teks	Hasil Cleaning
Salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini, disebabkan oleh perubahan dinamika kebutuhan masyarakat serta rendahnya tingkat supply akan barang.	Salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini disebabkan oleh perubahan dinamika kebutuhan masyarakat serta rendahnya tingkat supply akan barang.

Sehingga, permintaan akan melambung tinggi dengan persediaan yang terbatas.	Sehingga permintaan akan melambung tinggi dengan persediaan yang terbatas.

2.2.3 *Case Folding*

Case folding adalah proses mengubah teks menjadi karakter – karakter kecil dan hanya karakter dari huruf ‘a’ sampai ‘z’, angka dan tanda baca yang akan diterima. *Case folding* akan dilakukan karena dokumen mengandung berbagai variasi dan bentuk huruf. Variasi huruf harus diseragamkan untuk menghilangkan gangguan pada saat pengambilan informasi [14]. Berikut merupakan hasil dari *case folding* yang akan ditampilkan pada tabel 2.3 dibawah ini.

Tabel 2.3 Contoh Case Folding

Teks	Hasil Case Folding
Salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini disebabkan oleh perubahan dinamika kebutuhan masyarakat serta rendahnya tingkat supply akan barang	salah satu penyebab terjadinya krisis ekonomi yang terjadi belakangan ini disebabkan oleh perubahan dinamika kebutuhan masyarakat serta rendahnya tingkat supply akan barang.
Sehingga permintaan akan melambung tinggi dengan persediaan yang terbatas	sehingga permintaan akan melambung tinggi dengan persediaan yang terbatas.

2.2.4 *Stopword*

Stopword merupakan tahapan penghilangan kata yang tidak memiliki makna atau kurang relevan dan sering muncul pada sekumpulan kata – kata dengan panduan kamus. *Stopword* dapat berupa kata penghubung, kata ganti, preprosisi, dan lain – lain [14]. Berikut merupakan contoh dari *stopword* yang dapat dilihat pada tabel 2.4 dibawah ini.

Tabel 2.4 Contoh Stopword

Teks		Hasil Stopword	
salah	oleh	salah	oleh
satu	**perubahan**	satu	ubah
penyebab	dinamika	sebab	dinamika
terjadinya	**kebutuhan**	terjadi	butuh
krisis	masyarakat	krisis	masyarakat
ekonomi	serta	ekonomi	serta
yang	**rendahnya**	yang	rendah
terjadi	tingkat	terjadi	tingkat
belakangan	supply	belakang	supply
ini	akan	ini	akan
disebabkan	barang	sebab	barang

2.2.5 *Stemming*

Stemming adalah pengubahan kata – kata yang berimbuhan menjadi kata dasar pembentuknya dengan panduan kamus. Misalnya seperti kata "bekerja" akan diubah menjadi kata "kerja", kata "kebahagian" diubah menjadi kata "bahagia". [14].

2.3 *Noun Filtering*

Noun Filtering adalah tahapan dimana dilakukan proses penyaringan kata benda dari kumpulan hasi proses *preprocessing* sebelum masuk ke tahapan selanjutnya. Penyaringan kata benda dilakukan dengan cara melakukan pengecekan pada setiap kata yang bertipe benda.

2.4 *Word Sense Disambiguation*

Word sense disambiguation (WSD) merupakan sebuah proses mengidentifikasi makna antar beberapa kata yang digunakan pada kalimat tertentu. WSD digunakan untuk menangani kata – kata dengan tulisan yang sama namun sebenarnya memiliki makna yang berbeda. Proses WSD diawali dengan

membandingkan setiap term antar kalimat [11]. Sebuah proses disambiguasi sangat memerlukan kamus wordnet untuk menentukan makna dari kata yang memiliki ambiguitas.WSD memiliki dua varian tugas, yaitu sampel leksikal (ditargetkan) dan semua jenis kata. Sampel leksikal adalah sistem diperlukan untuk mendisambiguasikan suatu set kata target terbatas yang ada pada suatu kalimat. Sedangkan semua kata, sistem melakukan disambiguasi semua kata yang ada pada teks [5].

2.5 Algoritma *Lesk*

Algoritma *lesk* adalah algoritma yang digunakan untuk menghilangkan ambiguitas pada makna kata. Algoritma ini merupakan salah satu algoritma yang dapat menyelesaikan masalah pada *word sense disambiguation* dengan menggunakan kamus. Algoritma ini bekerja dengan cara mengambil kata – kata dari masing – masing tetangganya dari setiap *term* target. Setiap kata tetangganya akan dilakukan proses pengecekkan. Kemudian, makna dari kata yang telah di lakukan pengecekkan akan dihitung kemiripannya dengan setiap sinonim set (sinset) dari *term* target. Sinset dengan nilai kemiripan tertinggi, akan dipilih sebagai makna dari kata [11].

2.6 Algoritma Nazief & Adriani

Algoritma Nazief dan Adriani dikembangkan pertama kali oleh Bobby Nazief dan Mirna Adriani. Algoritma ini, berdasarkan pada aturan morfologi bahasa Indonesia yang luas, yang dikumpulkan menjadi satu grup dan di-enkapsulasi pada imbuhan/*affixes* yang diperbolehkan dan imbuhan/*affixes* yang tidak diperbolehkan. Algoritma ini menggunakan kamus kata dasar dan mendukung perekaman, yakni penyusunan kembali kata – kata yang mengalami proses *stemming* berlebih.

Langkah-langkah algoritma Nazief dan Adriani adalah:

1. Kata yang belum di-*stemming* dicari pada kamus, jika ditemukan, kata tersebut dianggap sebagai kata dasar yang benar dan algoritma dihentikan.
2. Hilangkan *Inflectional suffixes,* yaitu dengan menghilangkan *particle*(“-lah”,”-kah”,”-tah”, atau “-pun”). Kemudian hilangkan *inflectional*

possessive pronoun suffixes (“-ku”,”-mu” atau “-nya”). Cek kata di dalam kamus kata dasar, jika ditemukan, algoritma dihentikan. Jika tidak lanjut ke langkah 3.

3. Hapus Derivational Suffix (“-i” atau ”-an”,”). Jika kata ditemukan dalam kamus kata dasar, maka algoritma berhenti. Jika tidak, maka lanjut ke langkah 3a:
   a. Jika akhiran “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “- an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus Derivational Prefix (“be-”,”di-”,”ke-”,”me-”,”pe-“,”se-” dan “te-“). Jika kata yang didapat ditemukan didalam database kata dasar, maka proses dihentikan, jika tidak, maka lakukan recoding.

   Tahapan ini dihentikan jika memenuhi beberapa kondisi berikut:
   a. Terdapat kombinasi awalan dan akhiran yang tidak diijinkan
   b. Awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.
   c. Tiga awalan telah dihilangkan.
5. Jika semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus, maka algoritma ini mengembalikan kata yang asli sebelum dilakukan stemming [7].

2.7 Wordnet

Wordnet merupakan basis data leksikal yang menyediakan tempat penyimpanan bahasa Inggris. Wordnet didesain untuk menyediakan hubungan antara empat label kelas perkataan (*Parts of Speech*, POS) – *noun*, *verb*, *adjective* dan *adverb*. Unit terkecil WordNet adalah synset yang merepresentasikan makna spesifik sebuah kata. Ia mengandung kata, penjelasannya, dan synonym. Makna spesifik satu kata di satu label kelas POS disebut sense. Tiap-tiap sense sebuah

kata berada di dalam synset yang berbeda. Synset sebanding dengan sense, yaitu struktur yang mengandung sekumpulan term dengan makna synonym. Manfaat dari WordNet itu sendiri merupakan tempat dari basis pengetahuan untuk memberikan penjelasan yang lebih spesifik kepada penggunanya agak tidak mengalami kerancuan atau menjauh dari arti/makna kata sebenarnya.

2.8 Faktorisasi Matriks

Faktorisasi matriks merupakan proses pemecahan atau penguraian suatu matriks menjadi beberapa matriks. Matriks-matriks hasil faktorisasi biasanya memiliki struktur tertentu dimana membuat beberapa operasi akan menjadi lebih sederhana. Beberapa contoh matriks hasil faktorisasi adalah matriks triangular (segitiga atas, segitiga bawah, diagonal), matriks ortogonal atau matriks yang memiliki urutan yang lebih kecil.

Secara umum, metode faktorisasi matriks dibagi menjadi dua kelompok, yaitu *direct method* dan *approximation method*. *Direct method* merupakan teknik yang secara teori memberikan nilai eksak dengan jumlah langkah terbatas. Contohnya adalah faktorisasi LU, faktorisasi Cholesky, dan faktorisasi QR. Sedangkan *approximation method* menggunakan suatu perkiraan solusi awal dan dilanjutkan dengan iterasi yang memberikan solusi hasil lebih baik. Tujuan metode ini adalah untuk mendapatkan cara meminimalisir perbedaan antara solusi perkiraan dengan solusi eksak. Contoh *approximation method* adalah *single value decomposition, matrix factorization,* dan *non-negative matrix factorization* [10].

2.9 *Non-negative Matrix Factorization*

Non-negative Matrix Factorization (NMF) merupakan metode untuk mendekomposisi matrix *term-by-sentence* non-negatif A yang berukuran $m \times n$ menjadi 2 buah matriks, yaitu *Non-negative Semantic Feature Matrix* (NSFM), W, dengan ukuran $m \times r$, dan *Non-negative Semantic Variable Matrix* (NSVM), H, dengan ukuran $r \times n$. Matriks A merupakan matriks yang berisis bobot *term* dalam kalimat dan berukuran jumlah *term* (m) × jumlah kalimat (n). Metode dekomposisi dengan NMF dapat dinyatakan dalam persamaan 2.1 dibawah ini [10].

	$A \approx WH$	(2.1)

Proses untuk mencapai kondisi $A \approx WH$ dilakukan menggunakan rumus jarak *Euclidean* untuk menghitung jarak antara matriks A dengan perkalian antara matriks W dan H. Jarak *Euclidean* dihitung dengan menggunakan persamaan 2.2 berikut.

	$\|\|\|A - B\|\|^2 = \sum_{ij}(a_{ij} - b_{ij})^2$	(2.2)

Simpan nilai perkalian matriks W x H sebagai matriks V, dan nilai jarak *Euclidean* sebagai *cost*. Kemudian, nilai elemen matriks W dan H akan secara konstan diperbaharui untuk mencapai kondisi $A \approx WH$ menggunakan aturan *multiplicative update* pada persamaan berikut.

	$H = H\frac{(V^T V)}{(W^T WH)}$	(2.3)

	$W = W\frac{(VH^T)}{(WHH^T)}$	(2.4)

Setelah kondisi $A \approx WH$ tercapai, proses berikutnya dilakukan perhitungan untuk mendapatkan nilai *weight elemen* (Hi) pada matriks H menggunakan persamaan berikut.

	$weight(H_{i*}) = \frac{\sum_{q=1}^{n} H_{iq}}{\sum_{p=1}^{r}\sum_{q=1}^{n} H_{pq}})$	(2.5)

Keterangan:

n = jumlah kalimat

q = indeks kolom kalimat, dengan 1 < *q* < *n*

H_{iq} = elemen – elemen matriks *H* pada posisi baris *i* tertentu

p = indeks baris, dengan 1 < *p* < *r*

H_{pq} = elemen – elemen matriks *H* pada posisi (*p*,*q*) yang merupakan keseluruhan elemen pada matriks *H*

Kemudian proses ekstraksi kalimat dilakukan dengan cara mengambil kalimat dengan nilai tertinggi. Nilai tersebut diperoleh dari perhitungan *Generic*

Relevance of Sentence terhadap setiap kalimat dalam matriks H menggunakan persamaan berikut.

$$GRS_j = \sum_{i=1}^{r} (H_{ij}.weight(H_{i*})) \quad (2.6)$$

Keterangan:

r = jumlah baris pada matriks *H*

i = indeks baris, dengan 1 < *i* < *r*

H_{ij} = elemen matriks *H* pada posisi (*i*,*j*)

Nilai GRS_j menunjukkan skor untuk setiap vektor kolom ke-*j* pada matriks *H*. Sedangkan, nilai *weight* merupakan bobot untuk elemen (*i*,*j*) pada matriks *H* [10].

2.10 Evaluasi Peringkasan Matriks

Evaluasi peringkasan adalah suatu cara untuk mengetahui hasil terhadap ringkasan itu baik oleh peringkasan secara manual dan memiliki tingkat akurasi yang baik atau tidak untuk peringkasan teks. Evaluasi terhadap sebuah ringkasan merupakan kegiatan yang subjektif. Oleh karena itu, evaluasi sebuah ringkasan merupakan hal yang tidak mudah.

Evaluasi yang digunakan pada penelitian ini adalah bersifat intrinsik, yaitu pengevaluasian dengan cara membuat ringkasan yang ideal kemudian nantinya hasilnya akan dibandingkan dengan ringkasan sistem [4].

1. *Precision*

Nilai proporsi dari suatu set yang diperoleh dari informasi yang relevan [16]. Persamaan nilai *precision* ditunjukkan pada persamaan 2.7 berikut:

$$Precision(P) = \frac{tp}{(tp + fp)} \quad (2.5)$$

True positive (tp) merupakan kalimat yang ada di dalam ringkasan manual dan muncul dalam ringkasan sistem. *False positive (fp)* merupakan kalimat yang ada di dalam ringkasan manual tapi tidak muncul di dalam sistem.

2. *Recall*

Recall adalah nilai proporsi dari semua dokumen yang relevan yang di koleksi termasuk dari dokumen yang diperoleh [16]. Persamaan nilai *recall* ditunjukkan pada persamaan 2.8 berikut:

	$Recall(R) = \frac{tp}{(tp + fn)}$	(2.6)

False negative (fn) merupakan kalimat yang ada di dalam ringkasan manual tetapi tidak muncul dalam peringkasan sistem.

3. *F-Measure*

F-measure merupakan harmonic atau hubungan antara *recall* dan *precision* yang merepresentasikan akurasi sistem [16]. Persamaan *f-measure* ditunjukan pada persamaan 2.9 berikut:

	$F - Measure = \frac{2RP}{(P + R)}$	(2.7)

Keterangan:

P = Precision

R = Recall

4. ROUGE

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) adalah metode yang digunakan untuk mengukur kualitas dari sebuah ringkasan. ROGUE akan membandingkan antara rangkuman yang dihasilkan oleh sistem terhadap rangkuman ideal yang dibuat oleh pakar. Dalam hal ini, ringkasan yang dihasilkan oleh sistem disebut dengan kandidat ringkasan, sedangkan ringkasan yang digunakan sebagai rangkuman ideal atau *Human Gold Standart* (HGS). Pengukuran ROUGE didasarkan pada jumlah unit yang *overlap* dari tiap kata terhadap kandidat ringkasan dengan rangkuman ideal. Jenis pengukuran dengan menggunakan ROUGE ada beberapa macam. ROUGE-1 adalah jenis pengukuran yang akan digunakan dalam penelitian ini.

Pengukuran ROUGE-N didasarkan pada kemunculan secara statistik dari *n-gram*. Secara formal, ROUGE-N adalah nilai recall dari *n-gram* yang ada pada kandidat ringkasan terhadap rangkuman ideal. Pengukuran nilai ROUGE-N dapat

dihitung dengan menggunakan persamaan 2.10. Dimana *n* merepresentasikan panjang dari *n-gram*. Sedangkan, $count_{match}$ adalah jumlah *n-gram* yang sama antara *n-gram* dari ringkasan oleh sistem dengan *n-gram* yang ada pada ringkasan ideal. Dengan penyebut dari persamaan tersebut merupakan jumlah total *n-gram* yang ada pada ringkasan.

$$ROUGE - N = \frac{\sum_{S \in Summ_{ref}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S \in Summ_{ref}} \sum_{gram_n \in S} count(gram_n)} \quad (2.10)$$

Ketika rangkuman ideal ringkasan yang digunakan lebih dari satu, maka dihitung satu per satu nilai ROUGE-N antara ringkasan yang dihasilkan oleh sistem *s* terhadap setiap rangkuman ideal referensi r_i. Selanjutnya diambil nilai maksimal ROUGE-N dari pasangan ringkasan sistem dan rangkuman ideal dari ringkasan. Secara formal, perhitungan ROUGE-N multi rangkuman ideal dapat dilihat pada persamaan 2.11.

$$ROUGE - N_{multi} = argmax_i \, ROUGE - N(r_l, s) \quad (2.11)$$

Dalam penelitian ini, digunakan ROUGE-1. Hal ini berarti bahwa jumlah *n-gram* yang dibandingkan antara ringkasan sistem dengan rangkuman ideal berjumlah satu. Jika yang dibandingkan adalah kata-kata, maka ROUGE-1 membandingkan per satu kata pada ringkasan sistem dengan ringkasan rangkuman ideal, bukan berupa rangkaian kata.

2.11 Pemodelan

Pemodelan adalah proses pengembangan dari model abstrak dari sistem dimana setiap model akan memperlihatkan gambaran yang berbeda dari sistem. Pemodelan yang digunakan dalam penelitian adalah sebagai berikut.

2.11.1 *Flowchart*

Flowchart adalah bagan yang menunjukan alur dari program atau langkah prosedur sistem secara logis. Dalam menggambarkan langkah prosedur dalam sistem, digambarkan dengan simbol dan setiap simbol menyatakan gambaran dari

proses tertentu. Bagan alur program merupakan alur yang digunakan pada sistem [8].

2.11.2 Diagram Konteks

Diagram konteks merupakan diagram yang terdiri dari suatu proses dengan menggambarkan ruang lingkup dari suatu sistem. Diagram konteks menggambarkan bagian data flow diagram di level tertinggi yang menjelaskan alur dari input dan output sistem dan memberikan gambaran tentang keseluruhan sistem. Dalam diagram konteks, hanya ada satu proses [8].

2.11.3 Data Flow Diagram

Data flow diagram adalah diagram yang menggambarkan arus data sistem dengan menggunakan notasi simbol. Data flow diagram digunakan untuk menggambarkan secara logika dan menjelaskan arus data dari mulai masukkan sampai keluaran. Tingkat akan arus data mulai dari diagram konteks yang menjelaskan secara umum suatu sistem dari level 0 hingga dikembangkan menjadi level 1 sampai sistem tergambarkan seluruhnya. Gambaran ini tidak tergantung pada perangkat keras, perangkat lunak, struktur data atau organisasi file [8].

2.12 Bahasa Pemrograman

Bahasa pemrograman adalah instruksi atau aturan standar yang disusun sedemikian rupa sehingga memungkinkan pengguna untuk membuat program yang dapat dijalankan dengan instruksi – instruksti tersebut. Berikut merupakan beberapa contoh bahasa pemrograman, yaitu C, C++, C#, Java, JavaScript, PHP, Pytthon dan lainnya. Pada sistem peringkasan teks ini akan menggunakna bahasa pemrograman Phyton.

2.12.1 Phyton

Python adalah bahasa pemrogaman yang bersifat *open source*. Bahasa pemrograman ini dioptimalisasikan untuk *software quality*, *developer productivity*, *program portablitiy*, dan *component intregation*. Python telah digunakan untuk mengembangkan berbagai macam perangkat lunak, seperti

internet scripting, system programming, user interfaces, product customization, numberic programming dan lainnya [9].

2.13 Perangkat Lunak Pendukung

Perangkat lunak pendukung akan berguna sebagai perangkat lunak yang dapat mendukung dan mengembangkan pada aplikasi yang akan dibuat.

2.13.1 Sublime Text Editor

Sublime Text Editor merupakan salah satu teks editor untuk menyunting atau membuat sebuah aplikasi. Selain itu, mendukung pada sistem operasi seperti Windows, Linux dan Mac Os. Banyak fitur yang tersedia pada subltime text editor diantaranya, membuka script secara *side* to *side*, highlight serta lainnya. Perangkat lunak sublime text editor yang digunakan adalah yang memiliki versi 3.